

Проектная работа "Фракталы в
компьютерной графике" по
информатике.

Выполнил обучающийся 10 класса "А" Сидоренко В. А.

Руководитель Сеймовская М.Ю.

г. Ростов-на-Дону, 2021 год

Муниципальное бюджетное общеобразовательное учреждение г. Ростова-на-Дону «Школа № 107»
--

Содержание

1	Введение	3
2	Что такое фракталы?	3
3	Создание фракталов	5
3.1	Система повторяющихся функций	5
3.2	Аттрактор динамических систем	8
3.3	Система Линденмайера	9
3.4	Поточечное построение	12
3.5	Случайное построение	13
4	Фрактальное сжатие	14
4.1	Информация про сжатие	14
4.2	Алгоритм сжатие	16
4.3	Иные способы сжатия	17
5	Заключение	18

1 Введение

Поле компьютерной графики преодолевает много разных проблем в информационном веке. Одна из этих проблем - точная, реалистичная симуляция природных деталей. Полная реалистичность позволит создание полностью жизнеподобных ситуаций внутри компьютера. Для этого нужно симулировать случайность в развитии деталей природы. Другая из этих проблем - сжатие изображений. Каждое изображение построение из миллионов битов, каждый отвечающий на вопросы путем "да"или "нет описывающие цвет каждого пикселя. Хотя стоимость хранения одного бита крайне низка, хранить тысячи изображений может быть недешево. Нужно сжимать изображения, теряя небольшое количество качества ради меньшего количества необходимых битов. Фракталы могут быть ответом на оба из этих вопросов.

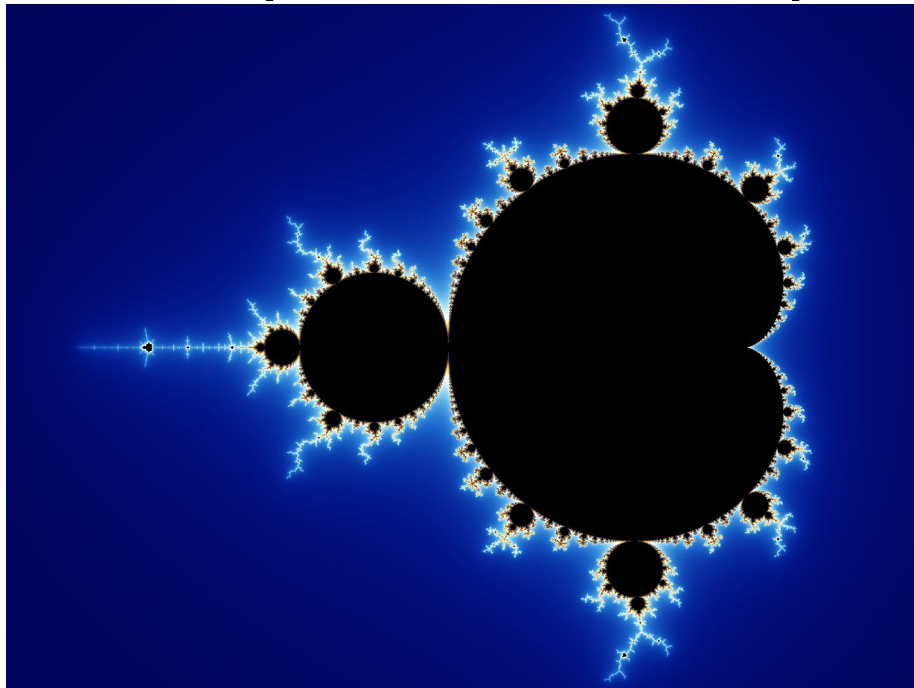
2 Что такое фракталы?

Математически, определение фракталов является спорным субъектом. Одни математики утверждают, что фракталы не должны иметь строгого определения. Можно использовать определение Фалконера. Фракталы должны иметь это:

- Самоподобность. Самоподобность не обязательно должна быть полностью точной. Множество Мандельброта - известный фрак-

тал, но его фрагменты не одинаковы друг к другу, а лишь похожи. Стохастично, случайно повторяющаяся последовательность также создает фрактал, что делает такие формы как берег Норвегии также фракталами.

Рис. 1: Раскрашенное множество Мандельброта



- Детальная структура формы в произвольно малых масштабах. Как ни приближаешься к фракталу, он никогда не становится похож на то, что построен из прямых линий.

Главное, что фракталы типично самоподобны. Не смотря, является ли самоподобность результатом функции или случайностью, оно останется фракталом.

3 Создание фракталов

3.1 Система повторяющихся функций

В создании большинства известных фракталов, используется система повторяющихся функций. Повторяющаяся функция - небесконечный набор сжимающих изображений на полном метрическом пространстве. Метрическим множеством называется множество, в котором между двумя точками определено расстояние. Обозначим расстояние между точками x и y как $p(x, y)$. Полным оно является если оно бесконечно во всех направлениях. Отрезок $(-\infty; \infty)$ является полным метрическим пространством, но не $(-\infty; 0)$. Сжимающее изображение - функция, изменяющая метрическое пространство и уменьшающая расстояние между любыми двумя выбранными точками. Функция $f(x)$ является сжимающим изображением, если для любых точек x и y в метрическом пространстве выполняется

$$p(f(x), f(y)) \leq \alpha * p(x, y)$$

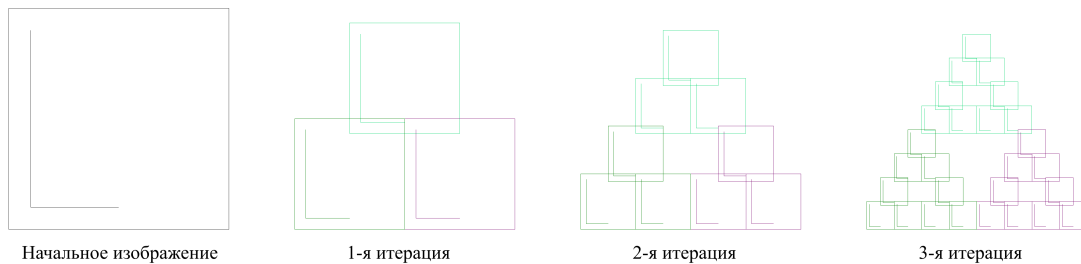
где $\alpha < 1$. При повторном выполнении сжимающего изображения на пространство снова и снова, оно будет стремиться к определенной точке, называемой аттрактором. Трансформации плоскости можно

выражать через произведение матриц:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$

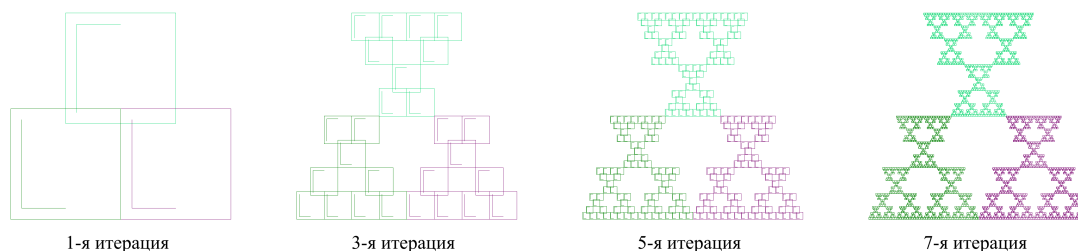
где a, b, c, d, e, f лежат в определении матрицы, а x и y - переменные, обозначающие позицию точки на плоскости. a по d отвечают за сжатие плоскости, так что $\begin{bmatrix} a & b \\ c & d \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a * x + b * y \\ c * x + d * y \end{bmatrix}$. Поскольку эта функция сжимающая, эти переменные лежат в промежутке $(-1; 1)$. e и f смещают получившееся сжатие на определенное расстояние на осях x и y соответственно. Типично, фракталы создаются системой повторяющихся функций, находя результаты каждой функции сначала на квадратной части плоскости с вершинами $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$ и $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$. В итоге создастся столько результатов, сколько и функций в системе. Затем находятся результаты использования системмы повторяющихся функций на полученные результаты, и это повторяется до бесконечности.

Рис. 2: Построение треугольника Серпинского системой из трех повторяющихся функций



В данном примере, используется система из 3 функций: $\begin{bmatrix} 0,5 & 0 \\ 0 & 0,5 \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix}$; $\begin{bmatrix} 0,5 & 0 \\ 0 & 0,5 \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0,5 \\ 0 \end{bmatrix}$; $\begin{bmatrix} 0,5 & 0 \\ 0 & 0,5 \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0,25 \\ \frac{\sqrt{3}}{4} \end{bmatrix}$ Изменение лишь одной из этих функций может привести к совсем другому фракталу. Как пример, можно изменить третью матрицу на $\begin{bmatrix} 0,5 & 0 \\ 0 & -0,5 \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0,25 \\ \frac{\sqrt{3}}{4} + \frac{1}{2} \end{bmatrix}$, что создаст верхний треугольник вверх ногами.

Рис. 3: Новая версия треугольника Серпинского



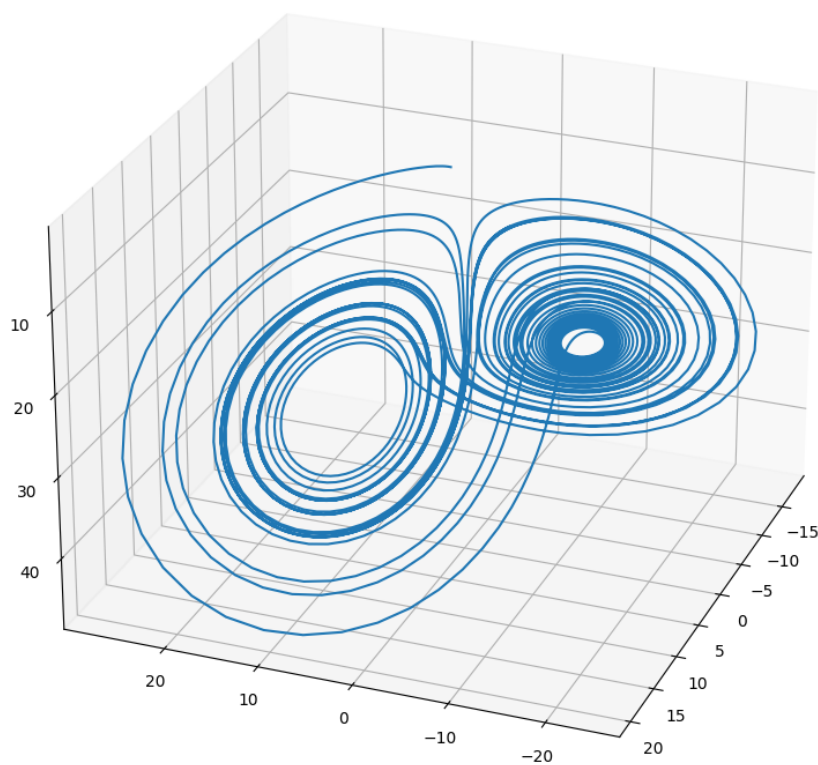
3.2 Аттрактор динамических систем

Иным способом является нахождение аттрактора определённых динамических систем. Динамические системы меняют свое значение со временем, и они могут стремиться к определенным значениям: точкам, серий точек, пути, или что-то другое - и это значение называется аттрактором системы. Если в итоге аттрактором является определенный путь, который никогда не пересекают любую данную точку дважды, то такой аттрактор называется "странным". Странные аттракторы являются фракталами. Примером такого фрактала является система Лоренца:

$$\begin{cases} \frac{dx}{dt} = \sigma(y - x) \\ \frac{dy}{dt} = x(\rho - z) - y \\ \frac{dz}{dt} = xy - \beta z \end{cases} \quad (1)$$

где σ, ρ, β - постоянны. Эта система имеет разные разные аттракторы в зависимости от значений постоянных и начальной точки, но при $\rho = 28, \sigma = 10, \beta = \frac{8}{3}$, большинство точек ведут к странному, фрактальному аттрактору. Уравнения Лоренца используются для симуляции гидродинамики, и фрактал - одно из возможностей этого.

Рис. 4: Аттрактор Лоренца, построенной использованием matplotlib в Python



3.3 Система Линденмайера

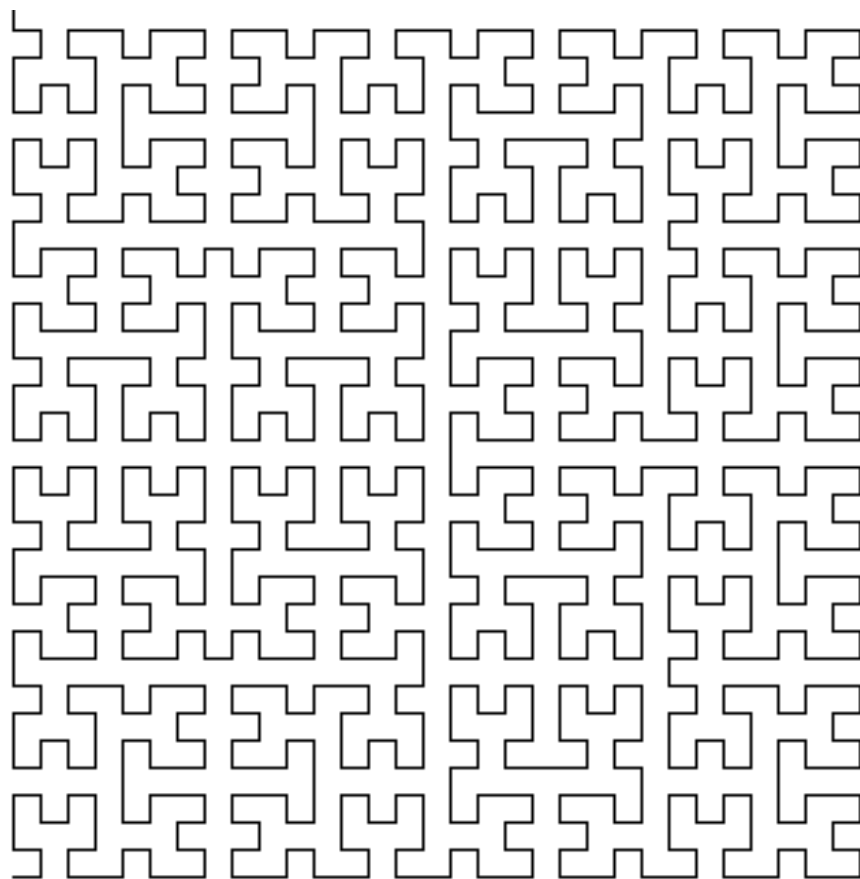
Для создания природных фракталов используется система Линденмайера. Это - система правил, позволяющая создавать фракталы выраженные как строка, и система состоит из 3 элементов: Алфавит, из которого будет составлена строка, аксиома - начальное значение строки до построения, и набор правил, диктующий изменение строки в каждой итерации.

Как пример можно привести типичную систему роста водорослей, с переменными A и B , аксиомой A и правилами $(A \rightarrow AB)$, $(B \rightarrow A)$. В итоге получается система:

$A; AB; ABA; ABAAB; ABAABABA; ABAABABAABAAB...$

Можно заметить, что количество букв в каждом слове следует последовательности Фибоначчи. Чаще всего, для визуализации используют отдельные буквы. Используя перо, можно иметь определенные буквы обозначать движение вперед пером, определенные обозначать поворот пути на кое-какое количество градусов, и т.д. С помощью системы Линденмайера определяются кривые Пеано, также известные как заполняющие пространство кривые, - прямые, проходящие через все точки единичного квадрата.

Рис. 5: Кривая Гильберта, построенная использованием turtle в Python

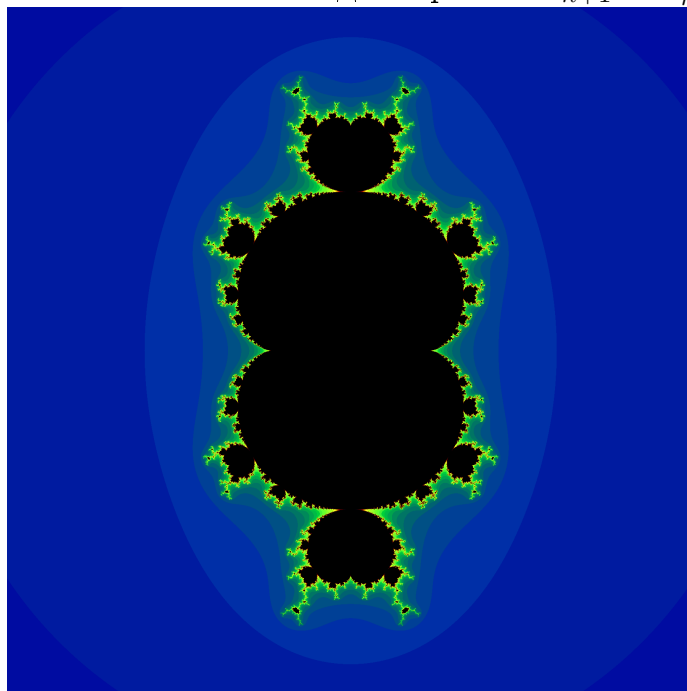


В типичной версии языка, аксиомы не зависят от контекста: одна буква превращается в комбинацию букв. В некоторых версиях системы, вводятся случайность или изменения, обращая внимание на окружающие буквы. Это позволяет создание большего количества результатов, которые могут быть более специализированными или случайными.

3.4 Поточечное построение

В иных случаях, каждая точка в фрактале в плоскости отдельно находится. Чаще всего, так обозначаются решения уравнений на комплексной числовой плоскости. Таким является множество Мандельброта - все значения c , для которых множество $z_0 = 0; z_{n+1} = z_n^2 + c$ является ограниченным. Конечно, c принимает значение $a + bi$ среди комплексных чисел, где $i = \sqrt{-1}$. Тогда $c^2 = a^2 - b^2 + 2abi$. Есть, конечно и иные версии множества Мандельброта, как при высших степенях.

Рис. 6: Множество Мандельброта с $z_{n+1} = z_n^3 + c$



3.5 Случайное построение

Иные фракталы имеют полностью случайное построение. Один из пример - построение Броуновского движения частиц в воде, основанное на уравнении Фоккера-Планка:

$$\frac{\partial W}{\partial t} = \left[- \sum_{i=1}^N \frac{\partial}{\partial x_i} D_i^1(x_1, \dots, x_N) + \sum_{i=1}^N \sum_{j=1}^N \frac{\partial^2}{\partial x_i \partial x_j} D_{ij}^2(x_1, \dots, x_N) \right] W$$

Это уравнение описывает шанс того, что скорость частицы лежит в интервале $(\mathbf{v}, \mathbf{v} + d\mathbf{v})$. В этом уравнении, D^1 - вектор перемещения, а D^2 - тензор диффузии. Полностью жизнеспособная симуляция жидкости до сих пор не достигнута в компьютерной графике, и эта область развивается.

4 Фрактальное сжатие

4.1 Информация про сжатие

Фрактальное сжатие основано на системе повторяющихся функций. Можно записать треугольник Серпинского как изображение, но меньше места будет занимать написание фрактала как систему из матриц. Если любую картину можно записать как несколько систем повторяющихся функций, то они будут занимать совсем мало места. На этом основан алгоритм фрактального сжатия.

По сравнению с другими способами сжатия изображений, фрактальное имеет преимущества. Фрактальное сжатие имеет очень высокий коэффициент сжатия - во сколько раз размер файла уменьшается по сравнению с несжатым изображением. Также получившееся изображение очень детально, если достаточно сделано итераций в расшивровании.

Но также есть и крупный минус: алгоритм сжатия и разжатия выполняется долго, гораздо медленнее чем такие алгоритмы как JPEG или PNG. Хотя и коэффициент сжатия лучше чем у JPEG, лишь на немного.

Теорема Банаха о неподвижной точке

Одна из причин фрактальное сжатие возможно - эта теорема. Для каждой функции $f(x)$, являющейся сжимающим изображением, есть одна и только одна точка, остающаяся неизменной при изменении, являющаяся аттрактором. Доказательство того, что нет не больше 1 точки очевидно: Если две точки остаются неизменными, то расстояние между ними неизменно, что противоречит определению сжимающего изображения. Доказательство того, что есть такая точка зависит от фундаментальности этой функции: сводится ли она к какому-нибудь значению. Возьмем x_0 как начальную точку, и $f(x_n) = x_{n+1}$. По определению сжимающего изображения, $p(x_1, x_2) \leq \alpha(x_0, x_1)$. Используя неравенство треугольника, $p(x_n, x_{n+p}) \leq p(x_n, x_{n+1}) + p(x_{n+1}, x_{n+2}) + \dots + p(x_{n+p-1}, x_{n+p}) \leq \alpha^n(1 + \alpha + \dots + \alpha^{p-1})p(x_0, x_1)$. Используя сумму геометрической прогрессии, получаем $\frac{\alpha^n - \alpha^{n+p}}{1 - \alpha} p(x_0, x_1)$. Так как по определению $0 < \alpha < 1$, $p(x_n, x_{n+p}) < \frac{\alpha^n}{1 - \alpha} p(x_0, x_1)$. Так что при $n \rightarrow \infty$, $p(x_n, x_{n+p}) \rightarrow 0$, обозначая то, что оно сходится к определенному числу.

Возьмем точку $x_t = \lim_{n \rightarrow \infty} x_n$. По неравенству треугольника, $p(x_t, x_{t+1}) \leq p(x_t, x_n) + d(x_n, x_{t+1}) = p(x_t, x_n) + d(x_n, x_{t+1}) \leq p(x_t, x_n) + \alpha d(x_{n-1}, x_t)$. По определению ограниченности, $p(x_t, x_n) < \epsilon$ и $\alpha d(x_{n-1}, x_t) < \epsilon$, где ϵ - произвольное положительное число. Значит, $p(x_t, x_{t+1}) < 2\epsilon$, и так как ϵ может быть любым положительным числом, $x_t = x_{t+1}$, что и следовало доказать.

4.2 Алгоритм сжатие

Главная часть фрактального сжатия изображений - нахождение подобных частей. Наиболее простое нахождение - с помощью прямоугольников со сторонами, параллельными осям координат. Однако экспериментируют с методом нахождения подобных частей треугольниками. Для нахождения подобности, полезно думать про изображение как про график функции. Для простоты мы представим черно-белое изображение 256 по 256 пикселей, к которому на каждый пиксель есть яркость с 256 возможными значениями. Это легко превратить в разноцветное изображение, представив его как 3 изображения в одном, выбирающие оттенки красного, зеленого и синего. Стоит отметить, что с добавлением яркости, матрица стала иметь 3 элемента вместо 2. Так что трансформирующая матрица имеет иной облик:

$$\begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & s \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e \\ f \\ o \end{bmatrix}$$

s умножает значение яркости на данное число, а o - добавляет это число к яркости.

Приведем пример с прямоугольниками. Возьмем два набора: R , состоящий из всех квадратов 8 на 8 пикселей, из которых сделано изображение, которые не могут пересекаться, и D , состоящий из всех квадратов 16 на 16 пикселей, из которых сделано изображение, ко-

торые могут пересекаться. В примере R будет иметь 1024 членов, а D - 58 081. Для каждого члена в первом наборе подставляются элементы второго набора и они сравниваются на подобность. Для этого используется средняя квадратическая формула в разницах пикселей - $\sqrt{\frac{a_1^2 + a_2^2 + \dots + a_n^2}{n}}$ - где n - количество членов. Так как D имеет в 4 раза больше пикселей, нужно уменьшить его размер - взять среднюю из каждого квадрата 2x2 или выбрать один пиксель из каждого такого квадрата. При минимизировании средней квадратической также находятся нужные значения o и s .

В практике, этот алгоритм работает отлично. При достаточном количестве итераций с пустого изображения, получается новое, весьма подобное начальному. Однако, такой способ не так хорошо покрывает точные детали в наборе R из-за определенного размера каждого квадрата в наборе.

4.3 Иные способы сжатия

Одним из способов утилизует дерево квадрантов. Если квадрат недостаточно подобен на 1 цвет, то он разбивается на 4 подобных. Однако, этот выбор не зависит от того, что в изображении. Так что создаются лишние квадраты, замедляя процесс.

Для этого можно использовать другой способ. Прямоугольники рекурсивно делятся на 2, отделяя куски полностью одинакового цвета или с диагонально идущим краем. Таким способом, создается меньше

прямоугольников, но в более важных местах, с большим самоподобием.

Иной способ использует треугольники, похоже на предыдущий способ. Сначала, прямоугольная картина разбивается на 2 треугольника, и затем, если выбран треугольник, то внутри него создается новый треугольник с произвольными точками на 3 сторонах. Одно из преимуществ лежит в том, что артефакты сжатия не имеют горизонтальные или вертикальные границы, так что они менее заметные.

5 Заключение

В итоге, фракталы имеют крупную практическую пользу в компьютерной графике. Фракталы лежат в основе реалистичных 3D моделей природы, в симуляции передвижения жидкости или иных физических явлений.

Но главная польза фракталов лежит в основе системы сжатия изображений. Новый, развивающийся алгоритм, основанный на самоповторении, имеет отличные результаты и может быть использован также, как и алгоритмами JPEG или PNG в будущем, с ростом силы процессоров.

Список литературы

- [1] Кеннет Фалконер. *Fractal geometry: mathematical foundations and applications*. (Английский)
- [2] Юваль Фишер. *Fractal image compression: theory and application* (Английский)

Список ресурсов

- <http://www.larryriddle.agnesscott.org/ifs/kit/index.htm> [Электронный ресурс] (Программа для построения фракталов системой повторяющихся функций)
- <https://stackoverflow.com/> [Электронный ресурс] (Сайт вопросов и ответов для программистов)